Web Search Using Automatic Classification

Chandra Chekuri
Michael H. Goldwasser
Computer Science Department, Stanford University.

Prabhakar Raghavan
Eli Upfal
IBM Almaden Research Center, 650 Harry Road, San Jose CA 95120.

Abstract:

We study the automatic classification of Web documents into pre-specified categories, with the objective of increasing the precision of Web search. We describe experiments in which we classify documents into high-level categories of the Yahoo! taxonomy, and a simple search architecture and implementation using this classification. The validation of our classification experiments offers interesting insights into the power of such automatic classification, as well as into the nature of Web content. Our research indicates that Web classification and search tools must compensate for artifices such as *Web spamming* that have resulted from the very existence of such tools.

Keywords:

Automatic classification, Web search tools, Web spamming, Yahoo! categories.

---

1. Introduction

Providing an efficient and user friendly search interface for the Web is still one of the most important challenges in making it accessible to the general public. Imagine that while researching cats, you wish to know the average weight of a jaguar. If you decide to search AltaVista with the keywords *jaguar* and *weight*, you will find roughly 900 matching documents, but unfortunately you will not immediately see any that answer your query. The results are clouded with many pages concerning Jaguar automobiles, the Atari Jaguar home game system, and possibly even the Jacksonville Jaguars football team. Of those 900, we have found that the highest document on the list that contains the information we want is document 41. (BTW, males average between 125-250 pounds.) Can we somehow tell a search engine (such as AltaVista) that our search using these keywords should be restricted to documents concerning zoology, or even just to science? One approach to restricting our search is to use a directory such as Yahoo!; unfortunately these (being manually generated) tend to cover only a small portion of the Web.

In fact, all currently available search tools suffer either from poor *precision* (i.e., too many irrelevant documents) or from poor *recall* (i.e., too little of the Web is covered by well-categorized directories). We address this by developing a search interface that relies on the automatic classification of Web pages. Our classification builds on the Yahoo! taxonomy, but differs in that it is automatic and thus capable of covering the whole Web substantially faster than the (human-generated) Yahoo! taxonomy. We describe experiments with our classifier; these tell us a great deal both about the particular classification implemented by Yahoo! as well as a great deal about the nature of Web content. In particular, we draw inferences on how the presence of search engines is influencing the content of Web in interesting ways that pose challenges to statistical classifiers such as ours, by studying the effect of

Web spamming [NYT96] on our classification.

1.1. Current Search Tools

Available search tools on the Web fall into two categories: *net directories* and *search engines*. Net directories, such as the one provided by Yahoo!, give a hierarchical classification of documents; each document in the directory is associated with a node of the tree (either a leaf or an internal node). Moving along the tree, a user can access a set of pages that have been manually pre-classified and placed in the tree.

Yahoo! for example consists today of a classification tree of depth of 10 or more (depending on the path followed). About 10-30 branches at each level of the tree lead to a total of a few hundreds of thousands of pages. Search in a net directory is very convenient and usually leads the user to the set of documents he is seeking, but it leads to only a small fraction of the Web (often the commercial part). This limited coverage stems from the (slow) rate of manual classification.

Search engines such as AltaVista and Excite cover a large portion of the Web. The drawback of these search engines is that they only support syntactic, keyword-oriented search, i.e., the search returns a list of pages that include a given set of keywords (or phrases). Most queries return either no page or a long list of pages, all of which include the given keywords, but most of which are irrelevant. The user must manually browse one document after another to find the page(s) sought. Some search engines offer "advanced" search features that enable Boolean combinations of search terms for improving the precision of the search. Aside from the limited improvement this can afford, one should not expect non-computer-literate users (whose ranks are growing) to be experienced at forming such Boolean formulae. (Note also that the "find similar pages" features in Excite and Infoseek require the user to first find at least one relevant page using syntactic search).

1.2. Information Retrieval

Note that the automatic classification of Web pages in our setting differs from the standard text classification problem studied in information retrieval (see, for instance, van Rijsbergen). There are two major differences: (1) For our purposes the classification does not need to be unique and always correct. Our automated classification of a document yields an ordered list of categories, ranked in order of likelihood. This information can significantly improve the search process even if the top category is not the "correct" one, or if the order of categories is not correct. (2) In traditional text classification, experimental validation is typically performed on structured corpora with well-controlled authoring styles (e.g., news articles). On the Web we do not have this luxury of a controlled style or quality of authorship. This poses interesting challenges as we show below.

1.3. Overview of this Paper

We describe a search interface that combines context-free syntactic search with context-sensitive search guided by classification, and that can potentially cover most of the Web. This allows the user to submit focused queries, thus improving on the precision of search engines while improving on the coverage of net directories.

To implement this idea we need an automatic process that accurately classifies Web pages. This process must be as efficient (in terms of time and space) as the process of building the index tables for the

syntactic search engine. We describe here a prototype that achieves these goals. We then describe experiments we performed to develop this prototype, the validation of the automatic classification, and the inferences one may draw from these experiments on the power of automatic classification and the Yahoo! directory.

In Section 2 we give a high-level description of the components of our search architecture. Section 3 describes the particulars of the experiments with Yahoo! used for our prototype. In Section 4 we draw inferences (based on our experiments) concerning the vocabulary of the Yahoo! taxonomy, and the effects of *Web spamming*, a direct result of the existence of Web search engines.

---

2. Architecture

2.1. The Automatic Classifier

Our classification process is statistical, and is based on term-frequency analysis. We begin with a set of categories and a pre-classified training set of pages. From these pages we build a (normalized) vector of frequencies of terms for each of the categories. One could obtain the training set from taxonomies like Yahoo! and Infoseek or from some other source depending on the classification desired. To classify a new document we compute the (normalized) word frequency vector of the document and compare it with the vectors representing the various categories. This comparison can be done using any of the several "similarity" or "distance" measures proposed in the information retrieval literature [SM83], [vanR79]. We output an ordered list of the document's most similar categories. Figure 1 gives a pictorial view of the process. The pre-computed category list for each page is stored as meta information in the index tables.

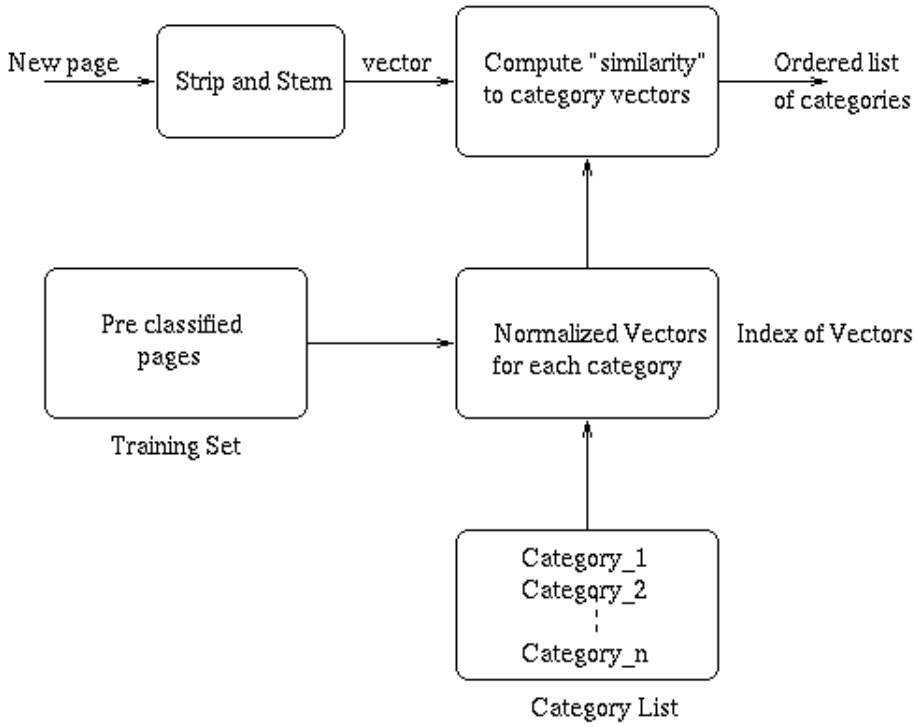

Figure 1 - Classification Overview

2.2. The Search Interface

We propose the following architecture for better search using automatic classification. The user specifies not only the keywords for a syntactic search, but also one or more categories about which he is interested. The search engine first retrieves the documents matching the keywords, then filters them by comparing their pre-computed categories against those chosen by the user. The filtering is a cutoff based process where a page matches the user's categories if any of them occurs within the first **k** categories of the ordered list assigned by the classifier to the page. The default value of **k** will be a small number which is a good compromise between precision and recall. Figure 2 gives an overview of the search architecture. One could use classification to aid the user's search in another way. For a given set of keywords, the search engine could return a list of categories under which the pages fall. This would let the user prune away unwanted pages without browsing through a long list. It could also make him aware of the kinds of ambiguities present in his keywords and prompt him to refine his search. To help the user in this, one could supplement the categories with keywords that identify the pages with their categories.

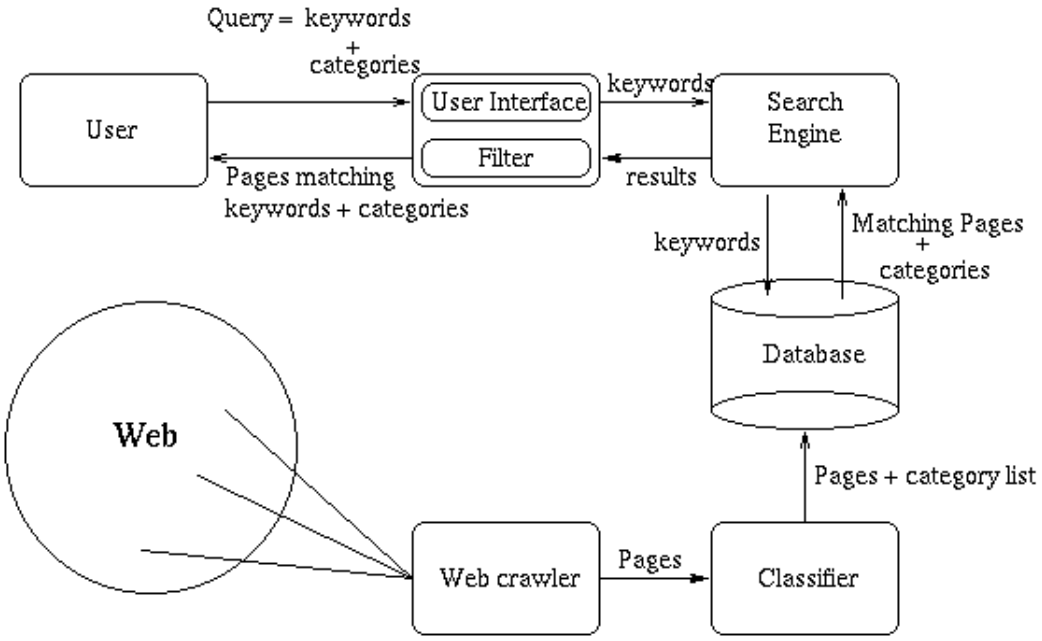

Figure 2 - Search Architecture

---

3. Validation Experiment

We now give details of the experiments with our classifier. We begin by describing the setup of our experiment, followed by the results. The experiments yield a number of surprising and interesting inferences about Web content.

3.1. Experiment Setup

We use a random sample of 2000 documents from each of the following 20 high-level Yahoo! categories to train our automatic classifier. Note that our categories do no exactly match the Yahoo! categories, as we chose to break several large topics into separate categories. Table 3 explains how we chose to form each of our categories from the top two levels of Yahoo!. After using these pages to train our classifier, we evaluated its performance by testing it against a new set of 500 randomly selected documents from each of the categories (*not including* documents used in the training phase). For both the training and classification we stripped the pages of their html tags but we did not *stem* the words. We believe that stemming will considerably improve the results. We used an inverted index to store the vectors for each category.

ID	Our Category	Relative to Yahoo!
Cp	Companies	Business_and_Economy:Companies, Business_and_Economy:Products_and_Services
Co	Computers	Computers_and_Internet (without Internet)
Ec	Economy	Business_and_Economy (without Companies, Products_and_Services)
Ed	Education	Education
FA	Fine_Arts	Arts (without Arts:Humanities)
Go	Government	Government
He	Health	Health
Hu	Humanities	Arts:Humanities
In	Internet	Computers_and_Internet:Internet
MT	Movies_TV	Entertainment:Movies_and_Films, News_and_Media:Television
Mu	Music	Entertainment:Music
NM	News_and_Media	News_and_Media (without Television)
Rc	Recreation	Entertainment, Recreation (without Movies_and_Films, Music, Sports)
RF	Regional_Foreign	Regional:Regions, Regional:Countries
RU	Regional_US	Regional:U.S._States
Re	Religion	Society_and_Culture:Religion
Sc	Science	Science
SS	Social_Science	Social_Science
So	Society_and_Culture	Society_and_Culture (without Religion)
Sp	Sports	Recreation:Sports

Table 3 - Our Twenty Categories

3.2. Experimental Results

Once trained, the classifier outputs an ordered list of possible categories for any given document. Assuming that the Yahoo! classification for that document is the "correct" one, we check the rank of that category in the list of categories generated by the classifier for that document. In Figure 4, we plot the recall percentage for various values of cutoff. That is, for a given cutoff value **k**, we assume that a document is correctly classified if its Yahoo! category is in the first **k** categories in the list. In more than 50% of the documents tested the Yahoo! classification came up first in the output of the automatic classifier, in more than 80% of the documents the Yahoo! classification was among the top 3 categories, and in more than 90% of the documents it was among the top 5 categories. Thus, for example, if a

typical document were associated with 5 categories, the syntactic search would focus on a smaller and more focused subset of the Web (in the absence of classification, we may instead think of all documents being classified under all 20 categories). These results indicate that our automatic classifier can enhance syntactic Web search; more on this below.

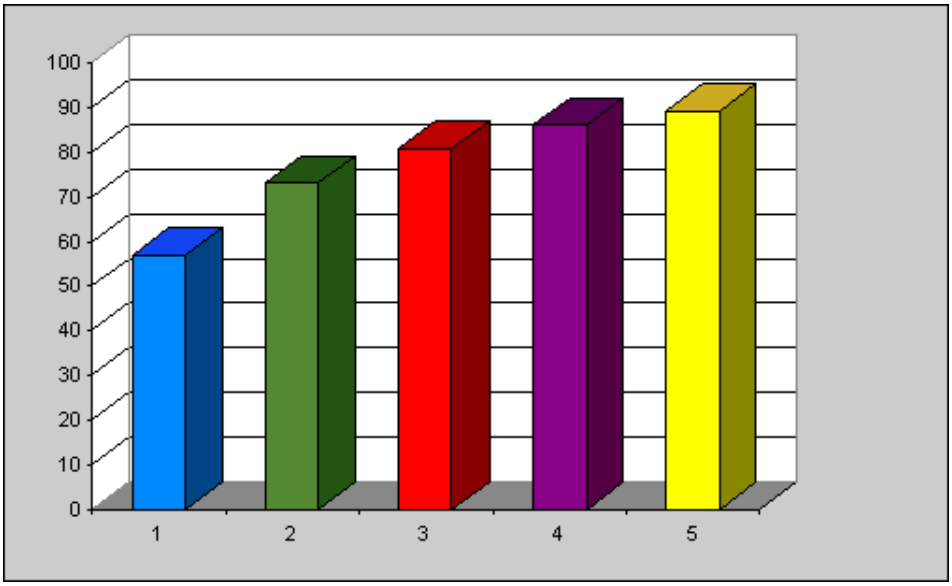

Figure 4 - Overall Recall Percentages for Cutoffs 1 to 5

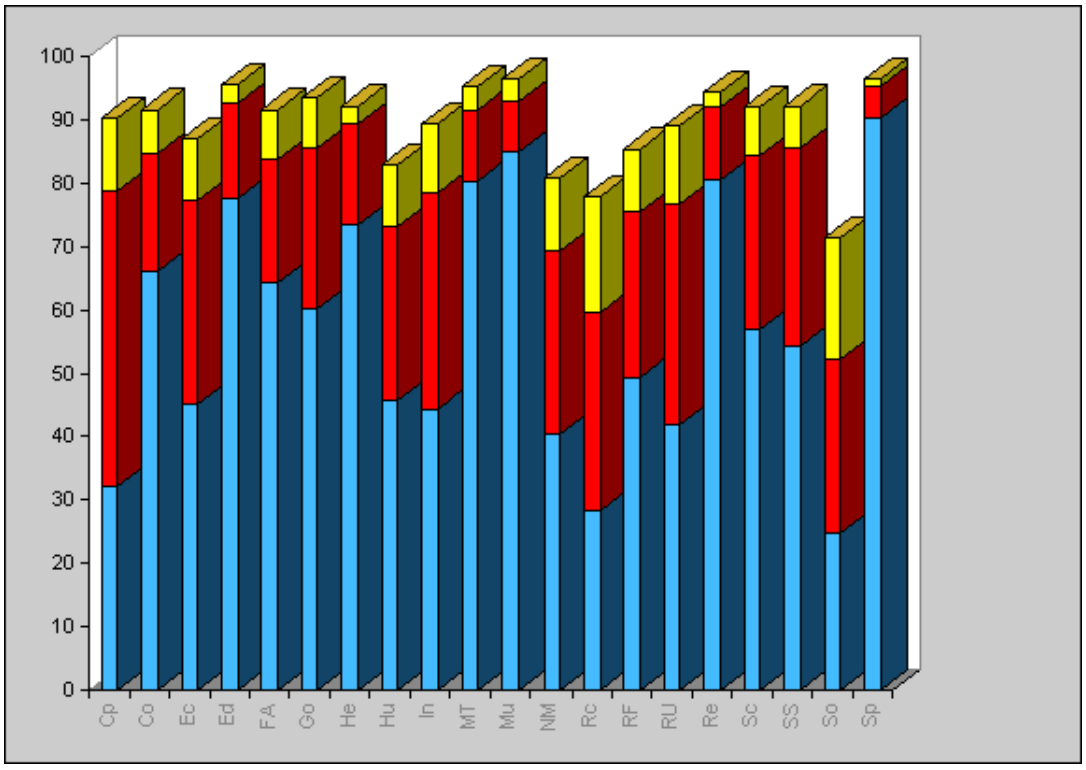

Figure 5 - Individual Category Recall Percentages for Cutoffs 1, 3 and 5

The quality of the classification process depends on the degree of "separation" between the categories used in the training process. This separation between categories depends on two factors:

- The ability of the training process to capture the differences between categories from a relatively small random sample;
- The inherent separation between the categories used.

Clearly, even the best training and classification process cannot achieve high precision when the categories are fuzzy and ambiguous. To further understand the outcome of our experiment we broke the precision results by categories. In the bar graph given in Figure 5, for each category we show the recall percentage for cutoff values of 1, 3 and 5 using a different color for the increments. Since the training and classification process was the same for all categories, the variance in precision between the categories has mainly to do with the characteristics of the different categories. Categories such as *Music* and *Sports* are well-defined, and there is little ambiguity in classifying documents into these categories. Other categories like *Society_and_Culture* and *Recreation* are not "well-separated" from other categories; thus classification into these categories is less clear, and changing the order between *Society* and *Recreation* in the list of categories for a document is not necessarily incorrect in some cases. The observation that many Web pages could be correctly classified under several categories strengthens our claim that Web documents should be classified into several possible categories, rather then uniquely classified (In fact, Yahoo! also employs many cross links between different places in their taxonomy, and so a given page may be accessible through several different high-level categories.) To further study the "separation" between the 20 Yahoo! categories we analyze the term-frequency vectors that result from our training process. We have computed the complete similarity matrix between categories, based on the similarity function used by the classification process. The graph in Figure 6 displays the relative similarities between the categories. For each category, we display edges to its two nearest neighboring categories. Additionally, we display the strength of the similarity by using wider edges for those relationships which are stronger.

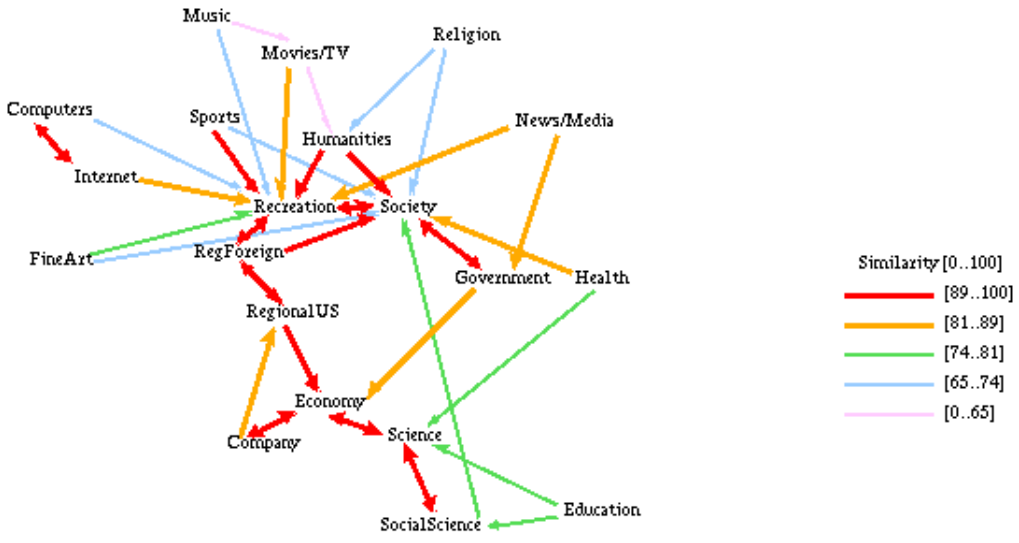

Figure 6 - Nearest Neighbors of our Categories

The graph clearly identifies categories that are well isolated from any other categories (e.g., *Music* and *Religion*) and pairs or groups of categories that are inherently close to each other and thus cause ambiguity in classification (*Computer* and *Internet*; *Companies* and *Economy*).

4. Category Vocabularies

Although our classifier was given no a priori information about the semantic content of the 20 categories, simply 2000 examples documents from each, its analysis of the term-frequencies offers us additional insight into the semantics of each category. After training our classifier, we are able to automatically produce a list of key words, which are the most distinguishing terms for each category. These terms are identified from the category's term-frequency vector, as words whose frequency is much greater in the category than in the corpus as a whole. Table 7 lists the top ten discriminating terms for each category, and the lists of the top 50 terms are found as hyperlinks from the table.

Category	Top Ten Most Discriminating Words
Companies	lessons, instruction, cars, driver, golf, photography, recording, classic, products, customers
Computers	linux, protocol, applet, bbs, os, modem, nt, shareware clock, comp
Economy	resume, jobs, models, fashion, ads, employment, consulting, marketing, investment, sales
Education	admissions, aid, loan, admission, financial, teachers, teacher, student, curriculum, learning
Fine_Arts	dance, theatre, artist, art, artists, architecture, photography, gallery, amateur, arts
Government	dole, republican, election, senate, democratic, campaign, vote, party, congress, gov
Health	patients, clinical, syndrome, cancer, therapy, surgery, disease, treatment, drug, medicine
Humanities	genealogy, novel, son, looked, anne, battle, poetry, fiction, war, texts
Internet	loser, irc, vrml, chat, cgi, translation, channel, domain, script, perl
Movies_TV	qv, jurassic, wars, trek, episode, movie, star, cast, film, hollywood
Music	jazz, album, guitar, band, bands, songs, concert, midi, song, blues
News_and_Media	cnn, clinton, fm, newspaper, radio, officials, trial, said, broadcast, mars
Recreation	oz, wine, cards, fish, game, glass, amateur, moon, magic, players
Regional_Foreign	india, kong, hotels, islands, hotel, russia, asia, bus, irish, tel
Regional_US	malls, hurricane, mn, homes, hawaii, breakfast, mexico, estate, carolina, il
Religion	christ, worship, bible, church, jesus, ministry, prayer, jewish, holy, faith
Science	maui, physics, dogs, dog, psychology, engineering, mathematics, surface, satellite, laboratory
Social_Science	adj, ion, anthropology, au, pl, economics, criminal, vs, studies, justice
Society_and_Culture	gay, lesbian, recipes, sexual, sex, gender, abuse, police, crime, lead
Sports	hockey, coach, olympic, baseball, league, football, teams, ball, team, sport

Table 7 - Top 10 Distinguishing Words

Examining the list of key words, we find many interesting entries. Our lack of stemming manifests itself, for instance with the occurence of both "admission" and "admissions" in *Education*, or "artist" and "artists" in *Fine_Arts*. We also notice several abbreviations, such as "il", "mn", and "tel". Many of these effects could be remedied with the use of stemming and dictionaries, however it is not so clear that all of

these should be eliminated. Clearly, many abbreviations such as "os," "nt," and "irc" offer important information to a classifier about the origin of a document. Stemming algorithms developed in information retrieval for more structured documents, like news and scholarly articles, need to be adapted to handle web documents.

Furthermore, we notice several anomalies. Why is the term "maui" the most distinguishing term in our list of keywords in the *Science* category? Why is the term "loser" the most distinguishing term in the *Internet* category? We next consider these and other artifacts resulting (in our opinion) from the great diversity of authorship and content on the web. There is a wide range of document length and quality on the web. Inevitably, certain measures must be taken to minimize the skew and bias in our classification procedure that a small number of documents may introduce. Variation in length is addressed by the following classical technique: the frequency of words in a given document is normalized to eliminate bias from documents that are substantially longer than the rest. However there are other, more pernicious sources of difficulty that a classifier used to support web search must be resilient to; curiously, these difficulties are a direct consequence of web search engines.

4.1. Web Spamming

A search engine based on keyword searches will often return thousands of documents that match a user's query. In an attempt to list the most relevant pages first, a search engine will often rely on a combination of heuristics, such as the number of occurrences of the search terms in the documents. For this reason, an increasingly popular technique used by designers of Web pages is to stuff a document with many repetitions of several key words. The goal of this technique is to exploit popular search engines so as to force their own page to jump towards the top of a possibly long list of query results. This technique was discussed in the **New York Times** [NYT96]. For example, if a consumer were interested in planning a vacation in Hawaii, and were to query AltaVista to find all documents that contain the words "maui" and "resort," she will find that over 40000 pages match the query. If we examine the page at the top of the list, we discover that the creators of this page have stuffed their page, placing many key words at both the beginning and end of the document, written using white letters on a white background.

In fact, up to 1000 of the 40000 Yahoo! pages that we chose for our training set seem to employ a variant of such Web spamming. For this reason, it is important to consider the effects of these pages on our automated classification. The effects of such pages can certainly be seen in our experiment. For example, in our list of words which most notably identify a page as "Science," we find that the top word is "maui." This seems surprising, and it turns out that our training set contains a page chosen from Science:Ecology at Yahoo!, titled "Maui Institute." This page simply discusses how nice the weather is every day in Maui, and hence the ontologists at Yahoo! agreed to list this page in its *Science* taxonomy. However, at the bottom of this page are 627 repeated occurrences of the words "maui golf real estate" making up 2500 of the 2700 words in the document, substantially affecting our classifier's view of science.

Not surprisingly, we found such techniques used commonly in commercial sites from our training set. In the *Company* category, we found single documents with repetitions of the following words: *pinball* (137 repetitions), *rolex* (308), *cars* (486), *skateboards* (525), *lighting* (540), and *golf* (1010). Another site stuffed their document with repetitions of several words, the only one of which we can print is *adult*. Furthermore, this technique was not limited to commercial sites. We found that a university department had placed 127 repetitions of their department name in the title of their page (they have since removed

them). Similarly, a medical school trauma center stuffed their page with 164 occurrences of the word *trauma*, comprising over 75% of the document. A student named *Sean*, apparently preparing for the job market, stuffed his page with 140 occurrences of his name, along with many other keywords identifying his field. Patriotism was big, with several tourism groups spamming the words: *Iran*, *India*, and *Kenya*. We found a bit of political activism, with 121 occurrences of the words *teen* and *protest* repeated in a page about teen curfews in Florida. Even the government chipped in, as a department of the National Weather Service decorated its page with 83 sprinkled occurrences of the word *snow*. The absolute winner in terms of quantity was the Relief Network page, devoted to helping us all recover from addictions, while habitually stuffing their own page with 2976 occurrences of the word *free*, along with several hundred occurrences of the words *smoking*, *drug*, *improvement*, *addiction* and others.

Although many such pages appeared in our training set, the results of our experiments reveal that the influence of such pages has a rather limited effect on our ability to classify other pages. The possible effects of such bias are discussed in Section 4.3, and methods for limiting such bias are suggested in Section 5.

4.2. Long Documents

It is not simply such Web spammers that pose a potential problem. Even legitimate pages may create artifacts if they are chosen in the random sample, as Web documents vary extraordinarily in length, and thus a category may be effected disproportionately by the vocabulary of a significantly long document. In fact, an astute reader will notice that in our experiment, the most distinguishing term in the *Internet* category was "loser" (no offense intended). The reason for this is that our training set contained LoserNet, a long story, chronicling "America's Favorite Loser," and containing the term "loser" 1532 times.

4.3. The Effects of Bias

In our experiments, we saw how lengthy individual pages in our random sample could bias our view of a category. These pages could have been the product of Web spammers, or simply disproportionately long but legitimate documents. As we saw, despite the bias caused by some individual pages in a category, our classification results were quite strong. The explanation for this robustness is quite clear. For example, we noticed that our categorizer was "tricked" into thinking that the term "maui" is highly correlated with the category *Science*. Therefore, if a new page contains the word "maui," that word may surely push for classifying the document as a *Science* page. However, when classifying a given document, each occurrence of a word in that document is given equal influence in deciding in which category this page is likely to belong. So a few occurrences of the word "maui", in a page with many other useful words, will be limited as the votes for all words are combined. It is likely that our results could be improved in several ways by limiting the effect of such biases. These issues are discussed in Section 5.

---

5. Further Work

We outline here a number of improvements that may lead to better classification.

As with any sampling-based technique it is important to tailor the sampling technique to minimize sampling error and bias. Two sources of bias already mentioned are: (1) very long documents; and (2)

spamming. The effect of long documents is controlled by measuring the frequency of a word in a document, rather than the number of times it appears in the document. Spamming can be controlled by limiting the ‘‘weight’’ of any single document in the outcome. A third source of bias is ‘‘missing documents’’. In collecting the sample documents one cannot ignore documents that are not accessible quickly (e.g., documents outside North America) when requested by the sampler. This could bias the sample, giving more weight to large, domestic sites. One can compensate for ‘‘missing documents’’ using standard missing-data sampling techniques.

As the precision of the classification improves one can aim at a more refined classification, discriminating between hundreds of different categories. As in Yahoo!, such a classification can be built hierarchically. Using stratified sampling a classifier can be trained for a large number of sub-classes using a relatively small number of sample documents.

As mentioned before, stemming can lead to better training and classification of documents. Scanning non-html documents (or parts of documents) such as imagemaps or frames for text can also help in classifying documents. Finally studying the links connected to and from a document could improve the accuracy of classifying that document.

Acknowledgments

We would like to thank Rob Barrett for providing us with most of the code for classification from the AIM prototype [KB95] and for his enthusiasm in helping us understand and debug it.

References

[FB92] *Information retrieval: data structures and algorithms.* William B. Frakes and Ricardo Baeza-Yates. Prentice Hall, Englewood Cliffs, N.J., 1992.

[KB95] *Subject-based searching using automatically extracted metadata - the AIM subject prototype.* T. Kirsche and R. Barrett. **IBM Research Report** , Oct. 27, 1995,

[NYT96] *Desperately Seeking Surfers; web programmers try to alter search engines' results,* Laurie Flynn, **New York Times**, Nov 11, 1996, p. C5.

[SM83] *Introduction to modern information retrieval.* Gerard Salton and Michael McGill. McGraw-Hill, New York 1983.

[vanR79] *Information Retrieval.* C.J. van Rijsbergen. Butterworths, London 1979.